



C++ - Módulo 00

Namespaces, classes, funções dos membros, stdio streams, listas de inicialização, estática, const, e algumas outras coisas básicas

Resumo:

Este documento contém os exercícios do Módulo 00 dos módulos C++.

Versão: 7

Conteúdos

I	Introdução	2
II	Regras gerais	3
III	Exercício 00: Megafone	5
IV	Exercício 01: A Minha Fantástica Lista Telefónica	6
V	Exercício 02: O Trabalho dos Seus Sonhos	8

Capítulo I

Introdução

C++ é uma linguagem de programação de uso geral criada por Bjarne Stroustrup como uma ex- tensão da linguagem de programação C, ou "C com Classes" (fonte: [Wikipedia](#)).

O objectivo destes módulos é apresentar-lhe a **Programação Orientada para Objectos**. Este será o ponto de partida da sua viagem em C++. Muitas línguas são recomendadas para aprender o OOP. Decidimos escolher C++ porque deriva do seu velho amigo C. Como se trata de uma linguagem complexa, e para manter as coisas simples, o seu código respeitará a norma C++98.

Estamos conscientes de que o C++ moderno é muito diferente em muitos aspectos. Portanto, se quiser tornar-se um desenvolvedor C++ proficiente, cabe-lhe a si ir mais longe depois do 42 Common Core!

Irá descobrir novos conceitos passo a passo. Os exercícios aumentarão progressivamente em complexidade.

Capítulo II

Regras gerais

Compilação

- Compilar o seu código com `c++` e as bandeiras `-Wall -Wextra -Werror`
- O seu código ainda deve ser compilado se adicionar a bandeira `-std=c++98`

Formatação e convenções de nomeação

- Os directórios de exercícios serão nomeados desta forma: `ex00`, `ex01`, ... , `exn`
- Nomeie os seus ficheiros, classes, funções, funções de membro e atributos, conforme exigido nas directrizes.
- Escrever os nomes das classes no formato **UpperCamelCase**. Os ficheiros contendo o código da classe serão sempre nomeados de acordo com o nome da classe. Por exemplo: `ClassName.hpp/ClassName.h`, `ClassName.cpp`, ou `ClassName.tpp`. Então, se tiver um ficheiro de cabeçalho contendo a definição de uma classe "BrickWall" que representa uma parede de tijolo, o seu nome será `BrickWall.hpp`.
- Salvo especificação em contrário, todas as mensagens de saída devem ser terminadas por um novo carácter de linha e apresentadas na saída padrão.
- *Adeus Norminette!* Nenhum estilo de codificação é aplicado nos módulos C++. Pode seguir o seu favorito. Mas tenha em mente que um código que os seus avaliadores não conseguem compreender é um código que eles não conseguem avaliar. Faça o seu melhor para escrever um código limpo e legível.

Permitido/Proibido

Já não está a codificar em C. Tempo para C++! Portanto:

- É-lhe permitido utilizar quase tudo da biblioteca padrão. Assim, em vez de se cingir ao que já sabe, seria inteligente utilizar o máximo possível as versões C++ish das funções C a que está habituado.
- No entanto, não se pode utilizar qualquer outra biblioteca externa. Isto significa que as bibliotecas C++11 (e formas derivadas) e Boost são proibidas. As seguintes funções são também proibidas: `*printf()`, `*alloc()` e `free()`. Se as usar, a sua nota será 0 e ponto final.

- Note que, salvo indicação explícita em contrário, a utilização do namespace `<ns_name>` e são proibidas as palavras-chave de amigos. Caso contrário, a sua nota será -42.
- **É-lhe permitido utilizar o STL apenas no Módulo 08.** Isto significa: sem **Containers** (vector/lista/mapa/e assim por diante) e sem **Algoritmos** (qualquer coisa que requeira incluir o `<algoritmo>` cabeçalho) até lá. Caso contrário, a sua nota será -42.

Alguns requisitos de concepção

- O vazamento de memória também ocorre em C++. Quando se atribui memória (utilizando o novo palavra-chave), deve evitar **fugas de memória**.
- Do Módulo 02 ao Módulo 08, as suas aulas devem ser concebidas na **Forma Canónica Ortodoxa**, excepto quando explicitamente indicado em contrário.
- Qualquer implementação de função colocada num ficheiro de cabeçalho (excepto para os modelos de funções) significa 0 para o exercício.
- Deverá poder utilizar cada um dos seus cabeçalhos independentemente dos outros. Assim, devem incluir todas as dependências de que necessitam. No entanto, deve evitar o problema da dupla inclusão, acrescentando **guardas de inclusão**. Caso contrário, a sua classificação será 0.

Leia-me

- Pode adicionar alguns ficheiros adicionais se precisar (ou seja, para dividir o seu código). Como estas atribuições não são verificadas por um programa, sintase à vontade para o fazer, desde que entregue os ficheiros obrigatórios.
- Por vezes, as directrizes de um exercício parecem curtas, mas os exemplos podem mostrar requisitos que não estão explicitamente escritos nas instruções.
- Leia cada módulo completamente antes de começar! A sério, fá-lo.
- Por Odin, por Thor! Usa o teu cérebro!!!



Terá de implementar muitas aulas. aborrecido, a menos que seja capaz de escrever o seu editor de texto favorito.


Isto pode parecer



É-lhe dada uma certa liberdade para completar os exercícios. No entanto, siga as regras obrigatórias e não seja preguiçoso. Iria perder muita informação útil! Não hesite em ler sobre conceitos teóricos.

Capítulo III

Exercício 00: Megafone

	Exercício : 00 Megafone
Anuário de entrada : <i>ex00/</i>	
Ficheiros para entregar : <i>Makefile, megafone.cpp</i>	
Funções proibidas : Nenhuma	

Só para garantir que todos estão acordados, escreva um programa que se comporte da seguinte forma:


```
$>./megafone "shhhhh... Acho que os alunos estão a dormir..."
SHHHHH... ACHO QUE OS ESTUDANTES ESTÃO A DORMIR...
$>./megaphone Damnit " ! " "Desculpem estudantes, pensei que esta coisa estava
desligada". DAMNIT ! DESCULPEM ESTUDANTES, PENSEI QUE ISTO ESTAVA DESLIGADO.
$>./megafone
* RUÍDO DE FEEDBACK ALTO E INSUPORTÁVEL *
$>
```



Resolver os exercícios de uma maneira C++.

Capítulo IV

Exercício 01: A Minha Fantástica Lista Telefónica

	Exercício : 01
	A minha lista telefónica fantástica
	Anuário de entrada : <i>ex01/</i>
	Ficheiros para entregar : Makefile, *.cpp, *.{h, hpp}
	Funções proibidas : Nenhuma

Bem-vindo aos anos 80 e à sua tecnologia inacreditável! Escreva um programa que se comporta como um software de lista telefónica fantástico.

É necessário implementar duas classes:

- **PhoneBook**

- Tem uma série de contactos.
- Pode armazenar um máximo de **8 contactos**. Se o utilizador tentar adicionar um 9º contacto, substituir o mais antigo pelo novo.
- É favor notar que a atribuição dinâmica é proibida.

- **Contacto**

- É sinónimo de um contacto telefónico.

No seu código, a lista telefónica deve ser instanciada como uma instância da classe da **PhoneBook**. O mesmo se aplica aos contactos. Cada um deles deve ser instanciado como uma instância da classe **Contactos**. É livre de conceber as classes como quiser, mas tenha em mente que tudo o que será sempre utilizado dentro de uma classe é privado, e que tudo o que pode ser utilizado fora de uma classe é público.



Não se esqueça de ver os vídeos da intranet.

No arranque do programa, a lista telefónica está vazia e o utilizador é solicitado a introduzir um de três comandos. O programa só aceita ADICIONAR, PESQUISAR e SAIR.

- **ADD:** guardar um novo contacto
 - Se o utilizador introduzir este comando, é-lhe pedido para introduzir a informação do novo campo de contacto um de cada vez. Uma vez preenchidos todos os campos, adicionar o contacto à lista telefónica.
 - Os campos de contacto são: nome, apelido, alcunha, número de telefone, e segredo mais obscuro. Um contacto guardado não pode ter campos vazios.
- **PESQUISA:** exibir um contacto específico
 - Mostrar os contactos guardados como uma lista de **4 colunas**: índice, nome próprio, apelido e apelido.
 - Cada coluna deve ter **10 caracteres** de largura. Um carácter de tubo ('\') separa-os. O texto tem de ser alinhado à direita. Se o texto for mais longo do que a coluna, deve ser truncado e o último caractere visível deve ser substituído por um ponto ('.').
 - Em seguida, solicitar novamente ao utilizador o índice da entrada a exibir. Se o índice estiver fora do intervalo ou errado, definir um comportamento relevante. Caso contrário, exibir a informação de contacto, um campo por linha.
- **SAIR**
 - O programa desiste e os contactos são perdidos para sempre!
- **Qualquer outra entrada é descartada.**

Depois de um comando ter sido correctamente executado, o programa espera por outro. Pára quando o utilizador introduz EXIT.


Dê um nome relevante ao seu executável.



<http://www.cplusplus.com/reference/string/string/> e, claro,
<http://www.cplusplus.com/reference/iomanip/>

Capítulo V

Exercício 02: O trabalho dos seus sonhos

	Exercício : 02
	O trabalho dos seus sonhos
	Directório de entrada : <i>ex02/</i>
	Ficheiros para entregar : <i>Makefile, Account.cpp, Account.hpp, testes.cpp</i>
	Funções proibidas : Nenhuma



Account.hpp, testes.cpp e o ficheiro de registo estão disponíveis para download na página da intranet do módulo.

Hoje é o seu primeiro dia na *GlobalBanksters United*. Depois de passar com sucesso nos testes de cruzeiro (graças a alguns truques do *Microsoft Office* que um amigo lhe mostrou), juntou-se à equipa de desenvolvimento. Sabe também que o recrutador ficou espantado com a rapidez com que instalou o *Adobe Reader*. Esse pequeno extra fez toda a diferença e ajudou-o a derrotar todos os seus adversários (também conhecidos como os outros candidatos): conseguiu!

De qualquer modo, o seu gerente apenas lhe deu algum trabalho a fazer. A sua primeira tarefa é recriar um ficheiro perdido. Algo correu mal e um ficheiro fonte foi apagado por engano. Infelizmente, os seus colegas não sabem o que é Git e usam chaves USB para partilhar o código. Nesta altura, faria sentido deixar este lugar agora mesmo. No entanto, decide-se ficar. Desafio aceite!

Os seus colegas desenvolvedores dão-lhe um monte de ficheiros. A compilação de testes.cpp revela que o ficheiro em falta é Account.cpp. Felizmente, o ficheiro de cabeçalho Account.hpp foi guardado. Existe também um ficheiro de registo. Talvez possa usá-lo para compreender como a classe **Account** foi implementada.

Começa-se a recriar o ficheiro Account.cpp. Em apenas alguns minutos, codifica algumas linhas de C++ puro e impressionante. Após algumas compilações falhadas, o seu programa passa nos testes. A sua saída corresponde perfeitamente à que foi guardada no ficheiro de registo (**excepto no que diz respeito aos carimbos temporais**, que obviamente serão diferentes, uma vez que os testes guardados no ficheiro de registo foram executados antes de ter sido contratado).

Raios, você é impressionante!



Pode passar este módulo sem fazer o exercício 02.

